

#nolist

/* DECLARATION FOR 'CODEMACRO' PART

Extended version of ASM86 */

dcl	comma	lit	','
	colon	lit	':'
	first	lit	'0'
	second	lit	'1'
	leftpar	lit	'{'
	rightpar	lit	'}'

#list

CP/M-86 v.1.1 (Vol. IV)

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # C81-162

```
$nolist
```

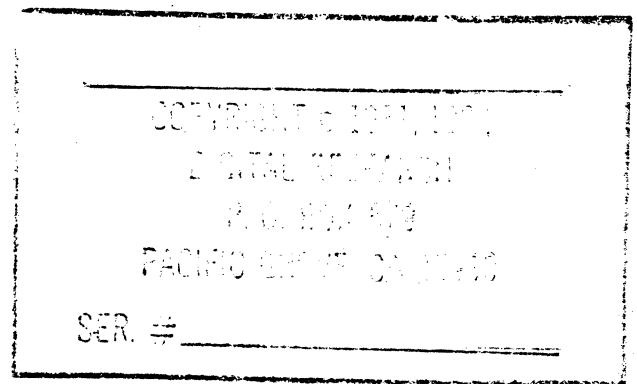
```
/* Literals used in codemacro specification : */
```

```
dcl
divisor lit '0',
multiplier lit '0',
place lit '0',
itype lit '0',
sifptr lit '0',
disptr lit '1',
port lit '1',
adr lit '0',
dst lit '0',
src lit '1',
opcode lit '0';
```

```
/* Literals to simplify table punching: */
```

```
dcl
cmachead lit 'struct (next address,nopar byte)',
cmac2struc lit 'cmachead,body(2) byte',
cmac3struc lit 'cmachead,body(3) byte',
cmac4struc lit 'cmachead,body(4) byte',
cmac5struc lit 'cmachead,body(5) byte',
cmac6struc lit 'cmachead,body(6) byte',
cmac7struc lit 'cmachead,body(7) byte',
cmac8struc lit 'cmachead,body(8) byte',
cmac9struc lit 'cmachead,body(9) byte',
cmac10struc lit 'cmachead,body(10) byte',
cmac11struc lit 'cmachead,body(11) byte',
cmac12struc lit 'cmachead,body(12) byte',
cmac13struc lit 'cmachead,body(13) byte',
cmac14struc lit 'cmachead,body(14) byte',
cmac15struc lit 'cmachead,body(15) byte',
cmac16struc lit 'cmachead,body(16) byte',
cmac17struc lit 'cmachead,body(17) byte',
cmac18struc lit 'cmachead,body(18) byte',
cmac19struc lit 'cmachead,body(19) byte',
cmac20struc lit 'cmachead,body(20) byte',
cmac21struc lit 'cmachead,body(21) byte';
```

```
$list
```



```
$nolist
$eject
```

```
/*      Here are the definitions for the      */
/*      codemacro instructions of the          */
/*      ASM86 assembler                        */
```

```
/* Commands within codemacros: */
```

```
declare
adbn      lit      '0',      /* DB with number */
adbf      lit      '1',      /* DB with formal parameter */
adwn      lit      '2',      /* DW with numbers */
adwf      lit      '3',      /* DW with formal parameter */
addf      lit      '4',      /* DD with formal parameter */
adbit     lit      '5',      /* DSIT */
aenda     lit      '6',      /* ENDM */
arelb     lit      '7',      /* RELB */
arelw     lit      '8',      /* RELW */
aendbit   lit      '9',      /* ENDBIT */
amodrm1   lit      '10',     /* MODRM with 1 formal parameter */
amodrm2   lit      '11',     /* MODRM with 2 formal parameters */
asegfix   lit      '12',     /* SEGFIX */
anosegfix lit      '13',     /* NOSEGFIX */
aformalbits lit      '14',   /* define bits from formal par. */
anumberbits lit      '15',   /* define bits from number */
```

```
/* Specifier letters: */
```

```
declare
specA     lit      '0',      /* accumulator, AX or AL */
specC     lit      '1',      /* code, address expression */
specD     lit      '2',      /* data, number used as immediate data */
specE     lit      '3',      /* effective address, either a memory
                               address (specH) or register (specR) */
specH     lit      '4',      /* memory address, variable (with or without
                               indexing) or [register expression] */
specR     lit      '5',      /* general register only (not segment) */
specS     lit      '6',      /* segment register */
specX     lit      '7',      /* simple variable name without indexing */
```

```
/* Modifier letters: */
```

```
declare
nomod     lit      '0',
modb      lit      '1',      /* byte expression */
modw      lit      '2',      /* word expression */
modsb     lit      '3',      /* byte in range (-128,127) */
modd      lit      '4',      /* 2-word expression */
```

```
/*      Segment override bytes: */
```

```
dcl
ESover lit '26h',
CSover lit '2eh',
SSover lit '36h',
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

DSover lit '3eh';

/* 'AND'-masks for codemaco head flag */

declare

no par\$and lit '0fh', /* no of parameters, bit 0-3 */
 prefix\$and lit '10h'; /* prefix flag, bit 4 */

/* 'OR'-masks for codemaco head flag */

declare

prefix\$on lit '10h'; /* PREFIX on flag */

/* 'AND'-masks for modifier-letter/range spec. byte */

declare

modletter\$bit lit '07h', /* bits 0-2 */
 range\$spec\$bit lit '0f8h', /* bits 3-7 */
 modletter\$count lit '0', /* bit position counters */
 range\$peccount lit '3';

/* 'OR'-masks for range-specifier bits */

declare

no range lit '0', /* no range specifier (bits 3-4) */
 single range lit '08h', /* single range */
 double range lit '10h', /* double range */
 range\$and lit '18h',
 number\$range lit '0', /* bit 5 */
 register\$range lit '20h',
 range\$type\$and lit '20h';

\$list

COPYRIGHT © 1981, 1982

SIGNAL RESEARCH

P.O. BOX 576

PACIFIC GROVE, CA 93950

SER. #

\$nolist

/* Special file devices if not diskfile : */

dcl

null	lit	'''Z'''-'A''', /* file devices */
printer	lit	'''Y'''-'A''',
console	lit	'''X'''-'A''',
validdisk	lit	'''P'''-'A''';

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
toolist
```

```
/*
```

```
modified 7/24/81 R. Silberstein
```

```
*/
```

```
/* Symbol types : */
```

```
dcl
reg      lit    '0',    /* register */
pseudo   lit    '1',    /* pseudo instruction */
code     lit    '2',    /* instruction */
string   lit    '3',    /* character string */
spec     lit    '4',    /* special character */
number   lit    '5',    /* 8 or 16 bit number */
variable lit    '6',
lab      lit    '7',    /* label */
operator lit    '8',    /* operator in expressions */
doubledf lit    '0f9h', /* doubled defined symbol */
neglected lit    '0fah', /* neglected symb., never to be def. */
ident    lit    '0fbh', /* identifier, scanner output */
error    lit    '0fch', /* error, scanner output */
undefsym lit    '0fdh', /* undefined symbol */
symbol   lit    '0feh', /* variable,label or undefined symb. */
deleted  lit    '0ffh', /* deleted symbol (not used) */
```

```
/* Symbol description values */
```

```
dcl
nil      lit    '0',    /* no specification */
byt      lit    '1',    /* symbol is 8-bit type */
wrđ      lit    '2',    /* symbol is 16 bit type */
dwrđ     lit    '4',    /* symbol is 2x16 bit type
                        or a segment register */
```

```
/* Register values : */
```

```
dcl
rax      lit    '0',    /* 16 bit registers */
rcx      lit    '1',
rdx      lit    '2',
rbx      lit    '3',
rsp      lit    '4',
rbp      lit    '5',
rsi      lit    '6',
rdi      lit    '7',

ral      lit    '0',    /* 8 bit registers */
rcl      lit    '1',
rdl      lit    '2',
rbl      lit    '3',
rah      lit    '4',
rch      lit    '5',
rdh      lit    '6',
rbh      lit    '7',
```

CONFIDENTIAL 101 1982
SUNFAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. =

```

res    lit    '0',    /* segment registers */
rcs    lit    '1',
rss    lit    '2',
rds    lit    '3';

```

```
/* Pseudo instructions: */
```

```

dcl
pob    lit    '0',
pdd    lit    '1',
pdw    lit    '2',
pif    lit    '3',
prs    lit    '4',
pend   lit    '5',
pequ   lit    '6',
porg   lit    '7',
pcseg   lit    '8',
pdbit   lit    '9',
pdseg   lit    '10',
pendm   lit    '11',
peseg   lit    '12',
prelb   lit    '13',
prelw   lit    '14',
psseg   lit    '15',
pendif   lit    '16',
pmodrm   lit    '17',
ptitle   lit    '18',
psegfix   lit    '19',
pinclue   lit    '20',
pject     lit    '21',
psimform   lit    '22',
pnosegfix   lit    '23',
ppagesize   lit    '24',
pcodemacro   lit    '25',
ppagewidth   lit    '26',
plist       lit    '27',
pnolist     lit    '28',
prb         lit    '29',    /* added in vers. 2.0 */
prw         lit    '30',
?IFLIST     LIT    '31',
PNOIFLIST   LIT    '32';

```

```
/* Symbolic operators */
```

```

dcl
oshort   lit    '0',    /* 8-bit value of expression */
oor       lit    '1',    /* logical OR */
oxor      lit    '2',    /* logical XOR */
oand      lit    '3',    /* logical AND */
onot      lit    '4',    /* logical NOT */
oeg       lit    '5',    /* equal */
ogt       lit    '6',    /* greater */
oge       lit    '7',    /* greater or equal */
olt       lit    '8',    /* less */

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

EQUALS.LIT;S

25-MAR-1982 20:20:41.40

Page 3

ole	lit '9',	/* less or equal */
one	lit '10',	/* not equal */
omod	lit '11',	/* arithmetic MOD */
oshl	lit '12',	/* shift left */
oshr	lit '13',	/* shift right */
optr	lit '14',	/* take type of 1. op, value of 2. */
ooffset	lit '15',	/* offset value of operand */
oseg	lit '16',	/* segment value of operand */
otype	lit '17',	/* type value of operand */
olength	lit '18',	/* length attribute of variables */
olast	lit '19',	/* length - 1 */
leftbracket	lit '['	
rightbracket	lit ']'	

*list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #


```
$nolist
```

```
/*
```

```
    modified 4/24/81 R. Silberstein
```

```
*/
```

```
/*
```

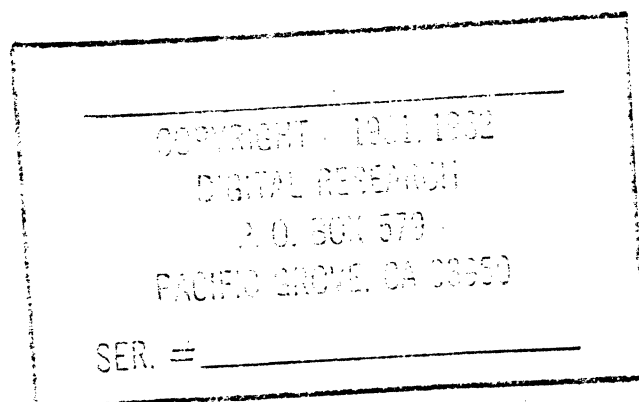
```
    This is all assembler error numbers.  
    For each error number there is a  
    corresponding error TEXT. The texts are  
    defined in the module ERMOD.PLM.
```

```
*/
```

```
dcl
```

firstitem	lit '0',	/* error in first item */
missingpseudo	lit '1',	
illegalpseudo	lit '2',	
doubledefvar	lit '3',	/* doubled defined errors: */
doubledeflab	lit '4',	
illegalmacro	lit '5',	/* illegal instruction name */
end\$of\$line\$err	lit '6',	/* garabage at end of line */
opmismatch	lit '7',	/* operands mismatch instruction */
illioper	lit '8',	/* illegal instruction operand */
missinstr	lit '9',	/* missing instruction */
undefsymbol	lit '10',	/* undefined element of expression */
pseudooperr	lit '11',	/* illegal pseudo operand */
nestediferr	lit '12',	/* nested IF illegal - ignored */
ifparerr	lit '13',	/* illegal IF operand - IF ignored */
missiferr	lit '14',	/* no matching 'IF' for 'ENDIF' */
neglecterr	lit '15',	/* neglected symbol */
doubledefsyab	lit '16',	/* doubled defined symbol */
instrerr	lit '17',	/* instruction not in code seqn. */
filesynterr	lit '18',	/* file name syntax error */
nestedincludeerr	lit '19',	/* nested INCLUDE not legal */
illexprelem	lit '20',	/* illegal expression element */
misstypeinfo	lit '21',	/* missing type info in operands */
laboutofrange	lit '22',	/* label out of range */
misseqminfo	lit '23',	/* missing segment info in operand */
codemacroerr	lit '24';	/* error in codemacrobuilding */

```
$list
```



\$olist
\$eject

/* PL/M language text macros: */

```

declare
lit      literally 'literally',
dcl      lit      'declare',
init     lit      'initial',
true     lit      'Offh',
false    lit      '0',
addr     lit      'address',
struc    lit      'structure',
proc     lit      'procedure',
reent    lit      'reentrant',
then$do  lit      'then do;',
else$do  lit      'end; else do;',
end$if   lit      'end',
forever  lit      'while true',
end$forever lit    'end',
end$while lit    'end',
end$case lit    'end',
end$do   lit    'end',
end$module lit    'end',
end$proc lit    'end',
cr       lit    '0dh',
lf       lit    '0ah',
tab      lit    '09h',
formfeed lit    '0ch',
end$of$file lit    'lah',
space    lit    '20h';

```

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

/* Convenient literals to compress source: */

declare

h literally 'structure(n address,1 byte,p address',

opcod2 literally 'h,a(2) byte)',

opcod3 literally 'h,a(3) byte)',

opcod4 literally 'h,a(4) byte)',

opcod5 literally 'h,a(5) byte)',

opcod6 literally 'h,a(6) byte)';

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P.O. BOX 570

RACING GROVE, CA 92650

SER. # _____

\$nolist

/* Output hex-record types: */

dcl

eoftype lit '01h', /* Common to both hex formats */
starttype lit '03h',

INTELdata lit '00h', /* INTEL hex format */
INTEL segment lit '02h',

DRcodedata lit '81h', /* Digital Research hex format */
DRdatadata lit '82h',
DRstackdata lit '83h',
DRextradata lit '84h',
DRcod-segm lit '85h',
DRdatasegm lit '86h',
DRstacksegm lit '87h',
DRextrasegm lit '88h',

CSdata lit '04h', /* Data types used inside assembler */
DSdata lit '05h',
SSdata lit '06h',
ESdata lit '07h',
CSvalue lit '08h',
DSvalue lit '09h',
SSvalue lit '0ah',
ESvalue lit '0bh';

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

/*

modified 3/27/81 R. Silberstein
modified 3/28/81 R. Silberstein
modified 4/13/81 R. Silberstein

*/

/* This file contains all structure definitions used */

dcl

file\$structure lit 'struct(
disk byte,
fcblock(33) byte,
bufptr addr,
buffer(512) byte)',

file\$structure lit 'struct(
disk byte,
fcblock(33) byte,
bufptr addr,
buffer(1024) byte)',

symbolstruc lit 'struct(
length addr,
stype byte,
sflag byte,
segment addr,
offset addr,
baseindex byte)',

operandstruc lit 'symbolstruc',

symbolhead lit 'struct(
next addr,
slength byte,
length addr,
stype byte,
sflag byte,
segment addr,
offset addr,
baseindex byte)',

codemacrohead lit 'struct(
next addr,
slength byte,
defptr addr)',

/* define bits of SFLAG of structures above */

type\$bit lit '7h', /* bit 0-2 */
segtype\$bit lit '13h', /* bit 3-4 */
seg\$bit lit '20h', /* bit 5 */
ireg\$bit lit '40h', /* bit 6 */
breg\$bit lit '80h', /* bit 7 */

COPYRIGHT - 1981, 1982
DIGITAL RESEARCH
P. O. BOX 379
PACIFIC GROVE, CA 93950
SER. #

```
/* left-shift counters */
typecount    lit '0',
segtypecount lit '3',
segmcount    lit '5',
iregcount    lit '6',
bregcount    lit '7',
```

```
/* define bits of BASEINDEX byte of structures above */
```

```
indexregbit  lit '01h', /* bit 0 */
baseregbit   lit '02h', /* bit 1 */
nooverridebit lit '40h', /* bit 6 */
```

```
/* left shift counters */
```

```
indexregcount lit '0',
baseregcount  lit '1',
noovercount   lit '6';
```

```
$list
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____